EXHIBIT F

```
Received: from arctic.openmarket.com (arctic.openmarket.com [199.170.183.8]) by relay.openmarket.com (8.6.10/8.6.6) with ESMTP
id LAA24620; Fri, 19 May 1995 11:53:59 -0400
Received: from OpenMarket.com (LOCALHOST [127.0.0.1]) by arctic.openmarket.com (8.6.10/8.6.6) with ESMTP id LAA20685; Fri, 19 M
ay 1995 11:53:58 -0400
Message-Id: <199505191553.LAA20685@arctic.openmarket.com>
X-Mailer: exmh version 1.6 4/21/95
To: billd@ai.mit.edu (Bill Dally)
cc: tml@OpenMarket.com, gifford@OpenMarket.com, stewart@OpenMarket.com,
        payne@OpenMarket.com, levergood@OpenMarket.com
Subject: Re: Session ID Patent
In-reply-to: Your message of "Fri, 19 May 1995 10:39:04 EDT."
             <9505191439.AA11072@grits>
Mime-Version: 1.0
Content-Type: text/plain; charset=us-ascii
Date: Fri, 19 May 1995 11:53:58 -0400
From: "Lawrence C. Stewart" <stewart@OpenMarket.com>
```

Just to contribute to the discussion!

I think the inventors are:
Levergood, Morris, Payne, Stewart, Treese

I am puzzled by the current discussion over the authentication step itself. The overall structure is:

An unauthenticated attempt to access protected content causes the content server to redirect the client to the authentication server, using a redirect URL of the form:

http://auth.server/auth-script?<originalURL+otherinfo, signed>

The authentication server then engages in a dialog with the client, to establish the user's access rights. This may include setting up an account registration right then and there.

The actual authentication dialog with the user may be many things, among which are the HTTP 1.0 "Basic Authentication" dialog from the HTTP spec.

It may also be the "Digest Authentication" scheme which is an IETF proposal, or a variety of other things.

The authentication dialog itself is not central to the SID ideas.

Once authenticated, the authentication server redirects the user back to the original URL, with a session ID included in the URL

The content server now recognizes the session ID and grants access.

I also wanted to raise a couple more issues, just in case we need to do anything to the application.

A

1) The netscape navigator browser has a cookie storage mechanism, in which one server can tell the browser "Store this cookie, and present it whenever you send a request to any server on this list".

This cookie mechanism would work for passing SIDs around. The auth server would set the cookie, and tell the browser to present it when any request is made to the content servers. This is just another way to store the SID. (We store it in the URL, which is universal, but also requires that the content be written with relative URLs.)

So the SID system can exploit browser specific features where they exist. The URL packaging is also not central other than making the system universal.

B

2) Do we describe the benefits of SIDs that go beyond access control? They include the ability for the content servers to be personalized, because the SID encodes the user ID on every hit. The content server can alter its behavior according to who is the user.

The SID can carry a payload of 96 bits in the current implementation, split among various fields. The number of bits can expand as necessary by making the SID longer. The information carried around can be expanded alongside.

-L